

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTT

```

EXE

Mod

EDT

ED

ED
EDED
EDED
EDED
ED

ED

ED

ED

ED

ED
EDSYN
LBA

110

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'EDT$KEYDEFKEY - interactive define key'
0002 0 MODULE EDT$KEYDEFKEY ( ! Interactive define key
0003 0 IDENT = 'V04-000' ! File: KEYDEFKEY.BLI Edit: JBS1017
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 Interactive define key
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: April 7, 1979
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 24-Feb-1981. This module was created by
0045 1 extracting routine EDT$SLN_DEFK from module KEYTRAN.
0046 1 1-002 - Regularize headers. JBS 09-Mar-1981
0047 1 1-003 - Use new message codes. JBS 04-Aug-1981
0048 1 1-004 - Add return values. JBS 02-Oct-1981
0049 1 1-005 - Revise call to enable autorepeat. JBS 30-Jan-1982
0050 1 1-006 - Make CR a valid part of the key definition. SMB 17-Feb-1982
0051 1 1-007 - Make DEFK illegal in NOKEYPAD mode - used only for defining
0052 1 keys in KEYPAD mode. SMB 1-Mar-1982
0053 1 1-008 - Fix routine header. JBS 22-Apr-1982
0054 1 1-009 - EDT$SDEFK returns a status. JBS 18-May-1982
0055 1 1-010 - Set a flag if control C actually aborts something. JBS 24-May-1982
0056 1 1-011 - Return true success from EDT$SNXT_CMDCH. STS 16-Jun-1982
0057 1 1-012 - Pass flag to EDT$SNXT_CMDCH to indicate whether or not we wish
```

EDTSKEYDEFKEY
V04-000

EDTSKEYDEFKEY - interactive define key

F 7
16-Sep-1984 00:42:57
14-Sep-1984 12:23:22

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]KEYDEFKEY.BLI;1

Page 2
(1)

```

: 58      0058 1 | to accept repeat counts. STS 16-Jun-1982
: 59      0059 1 | 1-013 - Modify message processing. SMB 22-Jun-1982
: 60      0060 1 | 1-014 - New implementation of defined keys. JBS 13-Aug-1982
: 61      0061 1 | 1-015 - Change cursor positioning call to obsolete a module. JBS 05-Oct-1982
: 62      0062 1 | 1-016 - Worry about having no cursor position to return to. This can happen
: 63      0063 1 | if this command is typed ahead initially. JBS 22-Feb-1983
: 64      0064 1 | 1-017 - Don't allow GOLD repeat count in key definitions. JBS 09-May-1983
: 65      0065 1 | --
: 66      0066 1 |
```


EDTSKEYDEFKEY
V04-000

EDTSKEYDEFKEY - interactive define key
Declarations

G 7
16-Sep-1984 00:42:57
14-Sep-1984 12:23:22

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]KEYDEFKEY.BLI;1

Page 3
(2)

```
: 68      0067 1 %SBTTL 'Declarations'
: 69      0068 1
: 70      0069 1 : TABLE OF CONTENTS:
: 71      0070 1 :
: 72      0071 1
: 73      0072 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 74      0511 1
: 75      0512 1 FORWARD ROUTINE
: 76      0513 1     EDT$LN_DEFK;
: 77      0514 1
: 78      0515 1 :
: 79      0516 1 : INCLUDE FILES:
: 80      0517 1 :
: 81      0518 1
: 82      0519 1 REQUIRE 'EDTSRC:EDTREQ';
: 83      0654 1
: 84      0655 1 LIBRARY 'EDTSRC:KEYPADDEF';
: 85      0656 1
: 86      0657 1 :
: 87      0658 1 : MACROS:
: 88      0659 1 :
: 89      0660 1 :     NONE
: 90      0661 1 :
: 91      0662 1 : EQUATED SYMBOLS:
: 92      0663 1 :
: 93      0664 1 :     NONE
: 94      0665 1 :
: 95      0666 1 : OWN STORAGE:
: 96      0667 1 :
: 97      0668 1 :     NONE
: 98      0669 1 :
: 99      0670 1 : EXTERNAL REFERENCES:
: 100     0671 1 :
: 101     0672 1 :     In the routine
```

! Interactive "define key"

EDT
V04

```
103 0673 1 %SBTTL 'EDT$$LN_DEFK - interactive define key'
104 0674 1
105 0675 1 GLOBAL ROUTINE EDT$$LN_DEFK ! Interactive define key
106 0676 1 =
107 0677 1
108 0678 1 ++
109 0679 1 FUNCTIONAL DESCRIPTION:
110 0680 1
111 0681 1 This routine is called when the change mode command DEFK is processed
112 0682 1 to process the interactive define key command. The user is prompted
113 0683 1 for the key to be defined and the defining string.
114 0684 1
115 0685 1 FORMAL PARAMETERS:
116 0686 1
117 0687 1 NONE
118 0688 1
119 0689 1 IMPLICIT INPUTS:
120 0690 1
121 0691 1 EDT$$T_CMD_BUF
122 0692 1 EDT$$G_CS [NO
123 0693 1 EDT$$G_LUR_COL
124 0694 1 EDT$$G_KPAB
125 0695 1
126 0696 1 IMPLICIT OUTPUTS:
127 0697 1
128 0698 1 EDT$$A_CMD_BUF
129 0699 1 EDT$$A_CMD_END
130 0700 1 EDT$$G_CC_DONE
131 0701 1
132 0702 1 ROUTINE VALUE:
133 0703 1
134 0704 1 1 = ok, 2 = end of journal file, 0 = fail or control C
135 0705 1
136 0706 1 SIDE EFFECTS:
137 0707 1
138 0708 1 May define a key, or undefine a key.
139 0709 1
140 0710 1 --
141 0711 1
142 0712 2 BEGIN
143 0713 2
144 0714 2 EXTERNAL ROUTINE
145 0715 2 EDT$$STOP_WKINGMSG,
146 0716 2 EDT$$CHK_CC,
147 0717 2 EDT$$SC_POSCISF : NOVALUE, ! Position the cursor
148 0718 2 EDT$$ERA_MSGLN,
149 0719 2 EDT$$MSG_BELL,
150 0720 2 EDT$$OUT_MSG,
151 0721 2 EDT$$TST_BADK,
152 0722 2 EDT$$DEFK,
153 0723 2 EDT$$CAN_KDEF,
154 0724 2 EDT$$NXT_CMDCH, ! Get the next command character
155 0725 2 EDT$$TRN_KSTR, ! Build the command buffer
156 0726 2 EDT$$PUT_CMDCH : NOVALUE, ! Put a character in the command buffer
157 0727 2 EDT$$SC_NONREVID,
158 0728 2 EDT$$TI_ENBLAUTREP, ! Enable or disable autorepeat
159 0729 2 EDT$$TI_DELK;
```



```
160 0730 2
161 0731 2
162 0732 2 EXTERNAL
163 0733 2 EDT$$G_MSGFLG,
164 0734 2 EDT$$G_MESSAGE_LINE,
165 0735 2 EDT$$G_KPAD,
166 0736 2 EDT$$T_CMD_BUF,
167 0737 2 EDT$$A_CMD_BUF,
168 0738 2 EDT$$G_CS [NO,
169 0739 2 EDT$$G_CUR_COL,
170 0740 2 EDT$$G_CC_DONE;
171 0741 2
172 0742 2 MESSAGES ((PRSKEYDEF, KEYNOTDEF, NOWENTDEF, BADDEFK));
173 0743 2
174 0744 2 LOCAL
175 0745 2 KEY_ADDR,
176 0746 2 KEY: INITIAL (K_KEY_MAX + 1),
177 0747 2 CON_U,
178 0748 2 C,
179 0749 2 SUCCEED,
180 0750 2 STATUS;
181 0751 2
182 0752 2 !+
183 0753 2 Since this command uses the bottom two lines for prompting and echoing,
184 0754 2 we don't want the Working message to continue printing.
185 0755 2 -
186 0756 2 EDT$$STOP_WKINGMSG ();
187 0757 2 !+
188 0758 2 The interactive define key facility only works in keypad mode.
189 0759 2 -
190 0760 2
191 0761 2 IF ( NOT .EDT$$G_KPAD)
192 0762 2 THEN
193 0763 2 BEGIN
194 0764 2 EDT$$MSG BELL (EDT$ BADDEFK);
195 0765 2 EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF);
196 0766 2 RETURN (0)
197 0767 2 END;
198 0768 2
199 0769 2 SUCCEED = 1;
200 0770 2 !+
201 0771 2 Set up the buffer for the definition to go into
202 0772 2 -
203 0773 2 EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
204 0774 2 EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF, 256);
205 0775 2 !+
206 0776 2 Prompt for the key.
207 0777 2 -
208 0778 2 EDT$$OUT_MSG (.EDT$$G_MESSAGE_LINE, EDT$_PRSKEYDEF, 0, 0);
209 0779 2 SUCCEED = EDT$$NXT_CMDCH (KEY, 0);
210 0780 2 ! zero means no repeat counts allowed
211 0781 2 IF (.SUCCEED NEQ 1) THEN RETURN (.SUCCEED);
212 0782 2
213 0783 2 !+
214 0784 2 Check for the validity of the key.
215 0785 2 -
216 0786 2
```

```
217 0787 2 IF EDT$STST_BADK (.KEY)
218 0788 2 THEN
219 0789 2 BEGIN
220 0790 2 !+
221 0791 2 Bad key. Put out a message and return.
222 0792 2 !-
223 0793 2 EDT$MSG_BELL (EDT$ KEYNOTDEF);
224 0794 2 EDT$A_CMD_END = CH$PTR (EDT$ST_CMD_BUF);
225 0795 2 RETURN (0);
226 0796 2 END;
227 0797 2
228 0798 2 !+
229 0799 2 Now prompt for the definition.
230 0800 2 !-
231 0801 2 EDT$OUT_MSG (.EDT$G_MESSAGE_LINE, EDT$NOWENTDEF, 0, 0);
232 0802 2 !+
233 0803 2 Loop, reading characters until we see the enter key. Keypad and function keys
234 0804 2 and gold/anything are translated, delete and CTRL/U have the usual
235 0805 2 meanings and all other characters are inserted directly.
236 0806 2 !-
237 0807 2 CON_U = 0; ! Note whether CONTROL U entered
238 0808 2
239 0809 2 WHILE 1 DO
240 0810 2 BEGIN
241 0811 2 SUCCEED = EDT$NXT_CMDCH (C, 0); ! No GOLD repeat counts
242 0812 2
243 0813 2 IF (.SUCCEED NEQ 1) THEN RETURN (2);
244 0814 2
245 0815 2 IF EDT$CHK_CC ()
246 0816 2 THEN
247 0817 2 BEGIN
248 0818 2 EDT$A_CMD_END = EDT$ST_CMD_BUF;
249 0819 2 EDT$G_CC_DONE = 1;
250 0820 2 RETURN (0);
251 0821 2 END;
252 0822 2
253 0823 2 SELECTONE .C OF
254 0824 2 SET
255 0825 2
256 0826 2 [K_ENTER] : ! The ENTER key, end of definition
257 0827 2 EXITLOOP;
258 0828 2
259 0829 2 [K_KPAD_BASE to K_KEY_MAX] :
260 0830 2 BEGIN
261 0831 2 !+
262 0832 2 Keypad, function or gold/something key, place the translation in the buffer.
263 0833 2 !-
264 0834 2 SUCCEED = EDT$TRN_KSTR (.C, 0, 1);
265 0835 2
266 0836 2 IF (.SUCCEED EQL 2) THEN RETURN (2);
267 0837 2
268 0838 2 EDT$A_CMD_END = CH$PTR (EDT$ST_CMD_BUF, 256);
269 0839 2 END;
270 0840 2
271 0841 2 [ASC_K_DEL] :
272 0842 2 BEGIN
273 0843 2 !+
```



```
274 0844 4 ! Delete key. Delete the previous character unless we are at
275 0845 4 ! the beginning.
276 0846 4 !-
277 0847 4
278 0848 5 IF CH$PTR_GTR (.EDT$$A_CMD_BUF, EDT$$T_CMD_BUF)
279 0849 4 THEN
280 0850 5 BEGIN
281 0851 5 EDT$$A_CMD_BUF = CH$PLUS (.EDT$$A_CMD_BUF, -1);
282 0852 5 EDT$$TI_DECK (CH$RCHAR (.EDT$$A_CMD_BUF));
283 0853 4 END;
284 0854 4
285 0855 3 END;
286 0856 3
287 0857 3 [ASC K_CTRL_U] :
288 0858 4 BEGIN
289 0859 4 !+
290 0860 4 ! CTRL/U: abort the define key command.
291 0861 4 !-
292 0862 4 CON U = 1;
293 0863 4 EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
294 0864 4 EXITLOOP;
295 0865 3 END;
296 0866 3
297 0867 3 [OTHERWISE] :
298 0868 3 !+
299 0869 3 ! None of the above: must be a character (other than DEL or CTRL/U) from
300 0870 3 ! the main keyboard. Insert it in the definition.
301 0871 3 !-
302 0872 3 EDT$$PUT_CMDCH (.C, 1);
303 0873 3 TES;
304 0874 3
305 0875 3 !+
306 0876 3 ! Turn on autorepeat, since seeing a keypad key will have turned
307 0877 3 ! it off.
308 0878 3 !-
309 0879 3 EDT$$TI_ENBLAUTREP (1);
310 0880 3 END;
311 0881 3
312 0882 3 !+
313 0883 3 ! Now define the key.
314 0884 3 !-
315 0885 3
316 0886 3 IF CH$PTR_NEQ (.EDT$$A_CMD_BUF, EDT$$T_CMD_BUF)
317 0887 3 THEN
318 0888 3 BEGIN
319 0889 3 STATUS = EDT$$DEFK (.KEY, EDT$$T_CMD_BUF, CH$DIFF (.EDT$$A_CMD_BUF, EDT$$T_CMD_BUF));
320 0890 3
321 0891 3 IF ( NOT .STATUS) THEN SUCCEED = 0;
322 0892 3
323 0893 3 END
324 0894 3 ELSE
325 0895 3 BEGIN
326 0896 3 !+
327 0897 3 ! Clear the definition of a key defined as the null string,
328 0898 3 ! except do nothing if the definition was terminated by a CTRL/U.
329 0899 3 !-
330 0900 3
```

EDT\$KEYDEFKEY
V04-000

EDT\$KEYDEFKEY - interactive define key
EDT\$\$LN_DEFK - interactive define key

L 7
16-Sep-1984 00:42:57
14-Sep-1984 12:23:22

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]KEYDEFKEY.BLI;1

Page 8
(3)

```
331 0901 4 IF (.CON_U EQL 0)
332 0902 3 THEN
333 0903 4 BEGIN
334 0904 4 STATUS = EDT$$CAN_KDEF (.KEY);
335 0905 4
336 0906 4 IF ( NOT .STATUS) THEN SUCCEED = 0;
337 0907 4
338 0908 4 END;
339 0909 4
340 0910 2 END;
341 0911 2
342 0912 2 !+
343 0913 2 Reset the buffer pointer, erase the command line and reposition the cursor if necessary
344 0914 2 before continuing.
345 0915 2 !-
346 0916 2 EDT$$A_CMD_BUF = EDT$$T_CMD_BUF;
347 0917 2 EDT$$A_CMD_END = EDT$$T_CMD_BUF;
348 0918 2 EDT$$G_MSGFLG = 1;
349 0919 2 EDT$$ERA_MSGLN ();
350 0920 2
351 0921 2 IF (.EDT$$G_CS_LNO GEQ 0) THEN EDT$$SC_POSCSIF (.EDT$$G_CS_LNO, .EDT$$G_CUR_COL);
352 0922 2
353 0923 2 EDT$$SC_NONREVID ();
354 0924 2 !+
355 0925 2 If we had an error print an appropriate message.
356 0926 2 !-
357 0927 2
358 0928 2 IF (.SUCCEED EQL 0) THEN EDT$$MSG_BELL (.STATUS);
359 0929 2
360 0930 2 RETURN (.SUCCEED);
361 0931 1 END;
```

! of routine EDT\$\$LN_DEFKEY

```
.TITLE EDT$KEYDEFKEY EDT$KEYDEFKEY - interactive defin
e key
.IDENT \V04-000\
.EXTRN EDT$$STOP WKINGMSG
.EXTRN EDT$$CHK_CC, EDT$$SC_POSCSIF
.EXTRN EDT$$ERA_MSGLN, EDT$$MSG_BELL
.EXTRN EDT$$OUT_MSG, EDT$$TST_BADK
.EXTRN EDT$$DEFK, EDT$$CAN_KDEF
.EXTRN EDT$$NXT_CMDCH, EDT$$TRN_KSTR
.EXTRN EDT$$PUT_CMDCH, EDT$$SC_NONREVID
.EXTRN EDT$$TI_ENBLAUTREP
.EXTRN EDT$$TI_DELK, EDT$$G_MSGFLG
.EXTRN EDT$$G_MESSAGE_LINE
.EXTRN EDT$$G_KPAD, EDT$$T_CMD_BUF
.EXTRN EDT$$A_CMD_BUF, EDT$$A_CMD_END
.EXTRN EDT$$G_CS_LNO, EDT$$G_CUR_COL
.EXTRN EDT$$G_CC_DONE, EDT$PRSKEYDEF
.EXTRN EDT$KEYNOTDEF, EDT$NOWENTDEF
.EXTRN EDT$BADDEFK
.PSECT _EDT$CODE,NOWRT, SHR, PIC,2
.ENTRY EDT$$LN_DEFK, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0675
```

OFFC 00000

5B	00000000G	00	9E	00002	MOVAB	R10,R11	
5A	00000000G	00	9E	00009	MOVAB	EDT\$MSG_BELL, R11	
59	00000000G	00	9E	00010	MOVAB	EDT\$NXT_CMDCH, R10	
58	00000000G	00	9E	00017	MOVAB	EDT\$OUT_MSG, R9	
57	00000000G	00	9E	0001E	MOVAB	EDT\$G_MESSAGE_LINE, R8	
56	00000000G	00	9E	00025	MOVAB	EDT\$A_CMD_END, R7	
55	00000000G	00	9E	0002C	MOVAB	EDT\$A_CMD_BUF, R6	
5E		04	C2	00033	MOVAB	EDT\$T_CMD_BUF, R5	
7E	03E8	8F	3C	00036	SUBL2	#4, SP	
00		00	FB	0003B	MOVZWL	#1000, KEY	0712
08	00000000G	00	E8	00042	CALLS	#0, EDT\$STOP_WKINGMSG	0756
	00000000G	8F	DD	00049	BLBS	EDT\$G_KPAD, T\$	0761
		3D	11	0004F	PUSHL	#EDT\$_BADDEFK	0764
53		01	DD	00051	BRB	3\$	
66		65	9E	00054	MOVL	#1, SUCCEED	0769
67	0100	C5	9E	00057	MOVAB	EDT\$T_CMD_BUF, EDT\$A_CMD_BUF	0773
		7E	7C	0005C	MOVAB	EDT\$T_CMD_BUF+256, EDT\$A_CMD_END	0774
	00000000G	8F	DD	0005E	CLRQ	-(SP)	0778
		68	DD	00064	PUSHL	#EDT\$_PRSKEYDEF	
69		04	FB	00066	PUSHL	EDT\$G_MESSAGE_LINE	
	04	7E	D4	00069	CALLS	#4, EDT\$OUT_MSG	
6A		AE	9F	0006B	CLRL	-(SP)	0779
53		02	FB	0006E	PUSHAB	KEY	
01		50	DD	00071	CALLS	#2, EDT\$NXT_CMDCH	
		53	D1	00074	MOVL	R0, SUCCEED	
		03	13	00077	CMPL	SUCCEED, #1	0781
		0146	31	00079	BEQL	2\$	
		6E	DD	0007C	BRW	19\$	
00000000G	00	01	FB	0007E	PUSHL	KEY	0787
0E		50	E9	00085	CALLS	#1, EDT\$TST_BADK	
	00000000G	8F	DD	00088	BLBC	R0, 4\$	
6B		01	FB	0008E	PUSHL	#EDT\$_KEYNOTDEF	0793
67		65	9E	00091	CALLS	#1, EDT\$MSG_BELL	
		33	11	00094	MOVAB	EDT\$T_CMD_BUF, EDT\$A_CMD_END	0794
		7E	7C	00096	BRB	6\$	0795
	00000000G	8F	DD	00098	CLRQ	-(SP)	0801
		68	DD	0009E	PUSHL	#EDT\$_NOWENTDEF	
69		04	FB	000A0	PUSHL	EDT\$G_MESSAGE_LINE	
		54	D4	000A3	CALLS	#4, EDT\$OUT_MSG	
		7E	D4	000A5	CLRL	CON U	0807
	08	AE	9F	000A7	CLRL	-(SP)	0811
6A		02	FB	000AA	PUSHAB	C	
53		50	DD	000AD	CALLS	#2, EDT\$NXT_CMDCH	
01		53	D1	000B0	MOVL	R0, SUCCEED	
		4B	12	000B3	CMPL	SUCCEED, #1	0813
00000000G	00	00	FB	000B5	BNEQ	8\$	
0D		50	E9	000BC	CALLS	#0, EDT\$CHK_CC	0815
67		65	9E	000BF	BLBC	R0, 7\$	
00000000G	00	01	DD	000C2	MOVAB	EDT\$T_CMD_BUF, EDT\$A_CMD_END	0818
		00FA	31	000C9	MOVL	#1, EDT\$G_CC_DONE	0819
	04	AE	DD	000CC	BRW	20\$	0820
00000141	8F	52	D1	000D0	MOVL	C, R2	0823
		78	13	000D7	CMPL	R2, #321	0826
0000012C	8F	52	D1	000D9	BEQL	14\$	
		29	19	000E0	CMPL	R2, #300	0829
000003E7	8F	52	D1	000E2	BLSS	10\$	
					CMPL	R2, #999	

EDTSKEYDEFKEY
V04-000

EDTSKEYDEFKEY - interactive define key
EDTSSLN_DEFK - interactive define key

N 7
16-Sep-1984 00:42:57
14-Sep-1984 12:23:22

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]KEYDEFKEY.BLI;1

Page 10
(3)

		20	14	000E9	BGTR	10\$	
		01	DD	000EB	PUSHL	#1	0834
		7E	D4	000ED	CLRL	-(SP)	
		52	DD	000EF	PUSHL	R2	
00000000G	00	03	FB	000F1	CALLS	#3, EDT\$\$TRN_KSTR	
	53	50	D0	000F8	MOVL	R0, SUCCEED	
	02	53	D1	000FB	CMPL	SUCCEED, #2	0836
		04	12	000FE	BNEQ	9\$	
	50	02	D0	00100	MOVL	#2, R0	
			04	00103	RET		
	67	0100	C5	9E 00104	MOVAB	EDT\$\$T_CMD_BUF+256, EDT\$\$A_CMD_END	0838
			3A	11 00109	BRB	13\$	0823
0000007F	8F	52	D1	0010B	CMPL	R2, #127	0841
		19	12	00112	BNEQ	11\$	
	50	65	9E	00114	MOVAB	EDT\$\$T_CMD_BUF, R0	0848
	50	66	D1	00117	CMPL	EDT\$\$A_CMD_BUF, R0	
		29	1B	0011A	BLEQU	13\$	
		66	D7	0011C	DECL	EDT\$\$A_CMD_BUF	0851
	50	66	D0	0011E	MOVL	EDT\$\$A_CMD_BUF, R0	0852
	7E	60	9A	00121	MOVZBL	(R0), -(SP)	
00000000G	00	01	FB	00124	CALLS	#1, EDT\$\$TI_DELK	
		18	11	0012B	BRB	13\$	0823
	15	52	D1	0012D	CMPL	R2, #21	0857
		08	12	00130	BNEQ	12\$	
	54	01	D0	00132	MOVL	#1, CON U	0862
	66	65	9E	00135	MOVAB	EDT\$\$T_CMD_BUF, EDT\$\$A_CMD_BUF	0863
		17	11	00138	BRB	14\$	0858
		01	DD	0013A	PUSHL	#1	0872
		52	DD	0013C	PUSHL	R2	
00000000G	00	02	FB	0013E	CALLS	#2, EDT\$\$PUT_CMDCH	
		01	DD	00145	PUSHL	#1	0879
00000000G	00	01	FB	00147	CALLS	#1, EDT\$\$TI_ENBLAUTREP	
		FF54	31	0014E	BRW	5\$	0809
	51	66	D0	00151	MOVL	EDT\$\$A_CMD_BUF, R1	0886
	50	65	9E	00154	MOVAB	EDT\$\$T_CMD_BUF, R0	
	50	51	D1	00157	CMPL	R1, R0	
		15	13	0015A	BEQL	15\$	
	50	65	9E	0015C	MOVAB	EDT\$\$T_CMD_BUF, R0	0889
7E	51	50	C3	0015F	SUBL3	R0, R1, -(SP)	
		55	DD	00163	PUSHL	R5	
		AE	DD	00165	PUSHL	KEY	
00000000G	00	03	FB	00168	CALLS	#3, EDT\$\$DEFK	
		0D	11	0016F	BRB	16\$	
		54	D5	00171	TSTL	CON_U	0901
		11	12	00173	BNEQ	17\$	
		6E	DD	00175	PUSHL	KEY	0904
00000000G	00	01	FB	00177	CALLS	#1, EDT\$\$CAN_KDEF	
	52	50	D0	0017E	MOVL	R0, STATUS	
	02	52	E8	00181	BLBS	STATUS, 17\$	0906
		53	D4	00184	CLRL	SUCCEED	
	66	65	9E	00186	MOVAB	EDT\$\$T_CMD_BUF, EDT\$\$A_CMD_BUF	0916
	67	65	9E	00189	MOVAB	EDT\$\$T_CMD_BUF, EDT\$\$A_CMD_END	0917
00000000G	00	01	D0	0018C	MOVL	#1, EDT\$\$G_MSGFLG	0918
00000000G	00	00	FB	00193	CALLS	#0, EDT\$\$ERA_MSGLN	0919
	50	00	D0	0019A	MOVL	EDT\$\$G_CS_LNO, R0	0921
		0F	19	001A1	BLSS	18\$	
	00000000G	00	DD	001A3	PUSHL	EDT\$\$G_CUR_COL	

EDT\$KEYDEFKEY
V04-000

EDT\$KEYDEFKEY - interactive define key
EDT\$\$LN_DEFK - interactive define key

B 8
16-Sep-1984 00:42:57
14-Sep-1984 12:23:22

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]KEYDEFKEY.BLI;1

Page 11
(3)

00000000G	00	50	DD	001A9	PUSHL	R0	
00000000G	00	02	FB	001AB	CALLS	#2, EDT\$\$SC_POSCSIF	
		00	FB	001B2	CALLS	#0, EDT\$\$SC_NONREVID	0923
		53	D5	001B9	TSTL	SUCCEED	0928
		05	12	001BB	BNEQ	19\$	
		52	DD	001BD	PUSHL	STATUS	
	6B	01	FB	001BF	CALLS	#1, EDT\$\$MSG_BELL	
	50	53	D0	001C2	MOVL	SUCCEED, R0	0930
			04	001C5	RET		
		50	D4	001C6	CLRL	R0	0931
			04	001C8	RET		

; Routine Size: 457 bytes, Routine Base: _EDT\$CODE + 0000

; 362 0932 1
; 363 0933 1 !<BLF/PAGE

EDT\$KEYDEFKEY
V04-000

EDT\$KEYDEFKEY - interactive define key
EDT\$SLN_DEFK - interactive define key

C 8
16-Sep-1984 00:42:57 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:23:22 [EDT.SRC]KEYDEFKEY.BLI;1

Page 12
(4)

: 365 0934 1 END
: 366 0935 1
: 367 0936 0 ELUDOM

! of module EDT\$KEYDEFKEY

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	457	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	5	1	40	00:00.2
\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
\$255\$DUA28:[EDT.SRC]KEYPADDEF.L32;1	34	3	8	7	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:KEYDEFKEY/OBJ=OBJ\$:KEYDEFKEY MSRC\$:KEYDEFKEY.BLI/UPDATE=(ENH\$:KEYDEFKEY)

: Size: 457 code + 0 data bytes
: Run Time: 00:20.7
: Elapsed Time: 00:26.7
: Lines/CPU Min: 2710
: Lexemes/CPU-Min: 8021
: Memory Used: 139 pages
: Compilation Complete

0135

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY